CLAIMS

1. A method for avoiding starvation of interrupts of a computer system, the computer system having at least one processor for servicing the interrupts, a memory subsystem, one or more input/output (I/O) devices configured to issue interrupts and at least one interrupt controller, the method comprising the steps of:

providing an interrupt queue register having a fixed number of entries for storing interrupts to be serviced by the at least one processor;

posting interrupts generated by the one or more I/O devices to the interrupt queue register;

determining whether the interrupt queue register is full and unable to store a new interrupt; and

if the interrupt queue register is full, one of waiting a preset time and reposting the new interrupt to the interrupt queue register, and draining the contents of the interrupt queue register into the memory subsystem so that the interrupt queue register can store the new interrupt.

2. The method of claim 1 further comprising the steps of:

if the interrupt queue register has a free entry for storing the new interrupt, sending an acknowledgement message (ACK) to the interrupt controller; and

if the interrupt queue register is full and cannot store the new interrupt, sending a non-acknowledgment message (NACK) to the interrupt controller.

3. The method of claim 2 further comprising the steps of:

providing an interrupt starvation counter that is operable by the interrupt controller;

in response to receiving an ACK from the processor, clearing the interrupt starvation counter; and

in response to receiving a NACK from the processor, adjusting the interrupt starvation counter.

4. The method of claim 3 wherein the step of adjusting comprises one of incrementing or decrementing the interrupt starvation counter.

5. The method of claim 3 further comprising the steps of:

determining whether the interrupt starvation counter exceeds a threshold; and

if the interrupt starvation counter does not exceed the threshold, performing the step of waiting the preset time and reposting the new interrupt to the interrupt queue register.

6. The method of claim 5 further comprising the steps of:

providing an interrupt request register having a plurality of bit positions that are settable to indicate a priority level of an interrupt posted to the processor;

if the interrupt starvation counter exceeds the threshold, setting the interrupt request register with a high priority indication; and

in response to the setting of the interrupt request register with the high priority indication, performing the step of draining the contents of the interrupt queue register into the memory subsystem.

7. The method of claim 6 further comprising the step of, in response to the posting of an interrupt to the interrupt queue register, setting the interrupt request register with a low priority indication.

8. The method of claim 7 wherein the interrupt request register includes a first bit segment the setting of which is indicative of a low priority interrupt and a second bit segment the setting of which is indicative of a high priority interrupt.

9. The method of claim 8 wherein each bit segment of the interrupt request register is associated with an interrupt line leading into the processor.

10. The method of claim 6 wherein the contents of the interrupt queue register drained into the memory subsystem are placed in a first-in-first-out (FIFO) queue.

11. The method of claim 10 further comprising the step of servicing the interrupts stored at the in-memory FIFO queue.

12. The method of claim 1 wherein the interrupts issued by the I/O devices are one of level sensitive interrupts (LSIs) and message signaled interrupts (MSIs).

13. The method of claim 11 wherein the interrupts issued by the I/O devices are one of level sensitive interrupts (LSIs) and message signaled interrupts (MSIs).

14. The method of claim 1 wherein the step of draining the contents of the interrupt queue register into the memory subsystem is performed by Privileged Architecture Library (PAL) code running at the processor.

15. The method of claim 1 wherein

the computer system includes a plurality of processors configured to issue interrupts, and

the interrupt controller receives interrupts from one or more selected processors of the computer system and posts such processor initiated interrupts to the interrupt queue of the at least one processor as with the I/O device interrupts.

16. A computer system comprising:

a plurality of input/output (I/O) devices configured to issue interrupts;

at least one processor for servicing the interrupts;

a memory subsystem;

an interrupt controller configured to receive the interrupts from the I/O devices and to post those interrupts to the at least one processor for servicing; and

an interrupt queue register for storing the interrupts posted to the at least one processor, wherein

upon receiving a newly posted interrupt, the at least one processor determines whether the interrupt queue register is already full and, if so, drains the contents of the interrupt queue register into the memory subsystem.

17. The computer system of claim 16 further comprising an interrupt starvation counter that is operable by the interrupt controller, wherein

if the interrupt queue register has a free entry for the new interrupt, sending an acknowledgement (ACK) message from the at least one processor to the interrupt controller, and

in response to the ACK, clearing the interrupt starvation counter.

18. The computer system of claim 17 further comprising

if the interrupt queue register does not have a free entry for the new interrupt, sending a non-acknowledgement (NACK) message from the at least one processor to the interrupt controller,

in response to the NACK, adjusting the interrupt starvation counter,

determining whether the interrupt starvation counter exceeds a threshold,

if the interrupt starvation counter does not exceed the threshold, waiting a predetermined time and reposting the new interrupt to the at least one processor.

19. The computer system of claim 18 further comprising an interrupt request register having a plurality of bit positions that are settable to indicate a priority level of an interrupt posted to the processor, wherein

if the interrupt starvation counter exceeds the threshold, setting the interrupt request register with a high priority indication; and

in response to the setting of the interrupt request register with the high priority indication, draining the contents of the interrupt queue register into the memory subsystem.

20. The computer system of claim 16 wherein

the computer system has a plurality of processors configured to issue interrupts, and

the interrupt controller posts the processor initiated interrupts to the at least one processor for servicing as with the I/O device interrupts.